Функции в R

Хотим сделать функцию, которая будет считать расстояние до точки в 2D или в 3D

```
dist_to_origin <- function(x){
    if (length(x) == 2) { # for 2D-case
        (x[1] * x[1] + x[2] * x[2]) ** 0.5
    } else if (length(x) == 3){ # for 3D-case
        (x[1] * x[1] + x[2] * x[2] + x[3] * x[3]) ** 0.5
    } else{
        stop("Can't calculate distance to origin")
    }
}
```

Если не тот размер, то ошибка

Минусы такого подхода

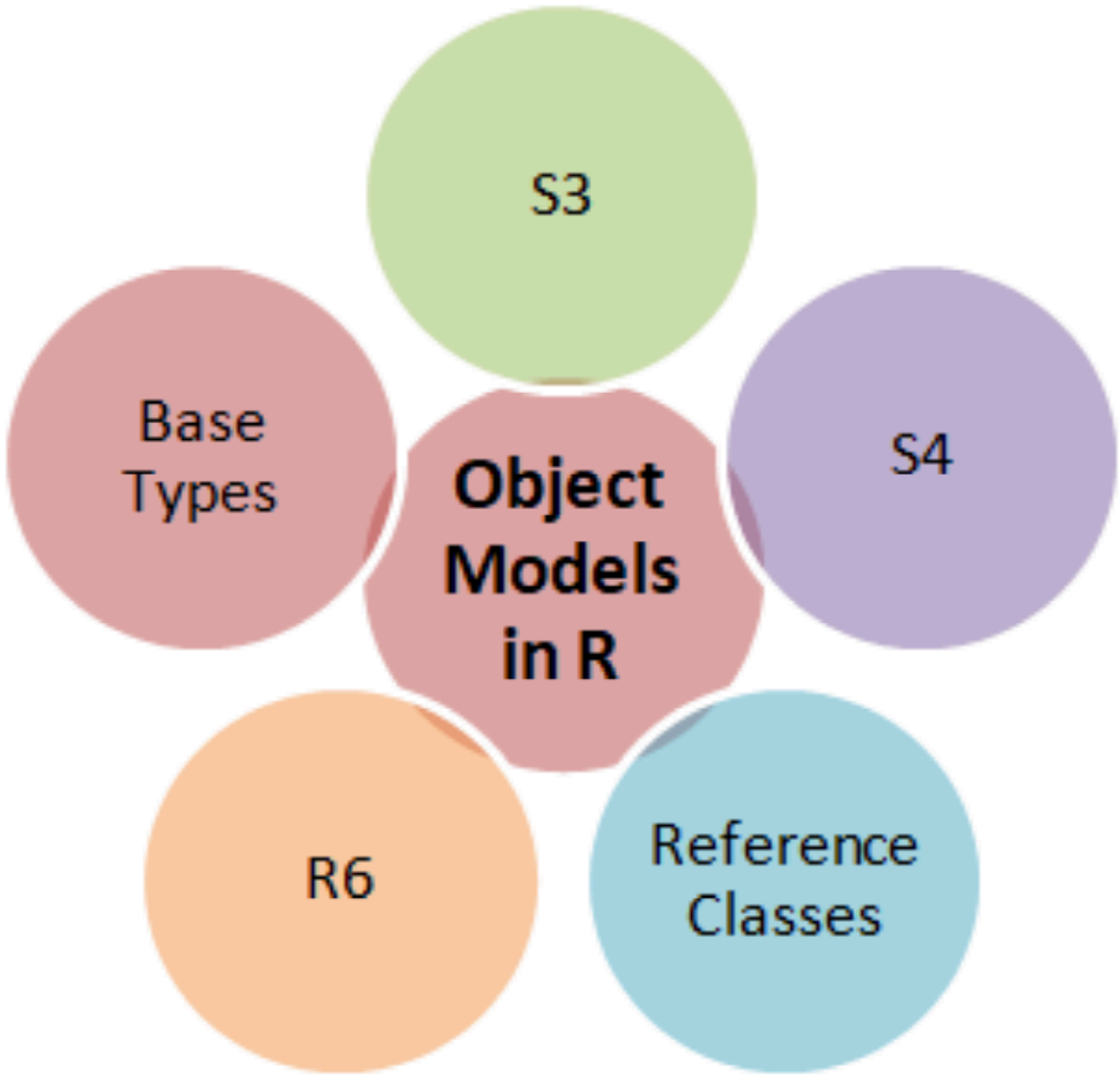
- Можем иметь несколько условий - на каждый набор условий нужно прописывать свое поведение;
- Можем иметь несколько схожих функций - в каждой функции надо писать один и тот же набор условий, можем ошибиться

ООП в R

Изначального ООП в R нет. Не подразумевалось, под него не закладывались какие-либо решения

Но ООП нужно, потому были сделаны надстройки. Так как надстройки получались с изюминкой, появилось 4 типа классов в R (ага, привет Perl)

ООП в R



Базовые типы

Узнаем при помощи команду typeof

```
library(rlang) # usually is not required
vec <- 1:10
print(typeof(vec))
```

```
## [1] "integer"
```

```
mat <- matrix(vec, nrow=2, ncol=5)
print(typeof(mat))
```

```
## [1] "integer"
```

```
fvec <- as.factor(vec)
print(typeof(fvec))
```

```
## [1] "integer"
```

```
l <- as.list(vec)
print(typeof(l))
```

```
## [1] "list"
```

```
df <- data.frame(a=vec)
print(typeof(df))
```

```
## [1] "list"
```

```
ht <- new.env(hash=TRUE)
print(typeof(ht))
```

```
## [1] "environment"
```

```
f <- function(){}
print(typeof(f))
```

```
## [1] "closure"
```

```
print(typeof(mean))
```

```
## [1] "closure"
```

```
print(typeof(rowMeans))
```

```
## [1] "closure"
```

```
ex <- expr(5 + 5)
typeof(ex)
```

```
## [1] "language"
```

```
qx <- quo(a+b)
typeof(qx)
```

```
## [1] "language"
```

S3-объекты

Самый малофункциональный способ. При этом самый простой и часто используемый

Создание объекта класса

```
x <- c(1, 2)
class(x) <- 'Point'
```

Создание объекта класса

Можно “сделать“ объект представителем нескольких классов

```
x <- c(1, 2)
class(x) <- c("A", "B")
```

Класс A “наследует” от B. Если нет метода для класса A, вызывается метод для класса B.

Методы

```
x <- c(1, 2)
class(x) <- 'Point2D'
y <- c(1, 2, 3)
class(y) <- 'Point3D'
```

Хотим написать функцию, возвращающую расстояние до начала координат для обоих классов

Методы

```
x <- c(1, 2)
class(x) <- 'Point2D'
dist_to_origin.Point2D <- function(x){
    (x[1] ** 2 + x[2] ** 2) ** 0.5
}

y <- c(1, 2, 3)
class(y) <- 'Point3D'
dist_to_origin.Point3D <- function(x){
    (x[1] ** 2 + x[2] ** 2 + x[3] ** 2) ** 0.5
}
```

Написали отдельно по функции для каждого из классов

Если бы не было нормального решения

```
dist_to_origin <- function(x) {
    if ('Point2D' %in% class(x)){
        dist_to_origin.Point2D(x)
    } else if ('Point3D' %in% class(x)){
        dist_to_origin.Point3D(x)
    } else{
        stop("no applicable method")
    }
}
```

Работает, но это боль.
И почти ничем не отличается от решения без классов.

Методы

Можем написать специальную generic function

```
dist_to_origin <- function(x, ...){
UseMethod("dist_to_origin", x)
}
x <- c(1, 2)
class(x) <- "Point2D"
print(dist_to_origin(x))
```

```
## [1] 2.236068
```

```
y <- c(1, 2, 3)
class(y) <- "Point3D"
print(dist_to_origin(y))
```

```
## [1] 3.741657
```

Работает.
Если дать класс, для которого нет соответствующей функции - dist_to_origin.class_name, то выдастся ошибка

Методы

```
y <- c(1, 2, 3)
class(y) <- "Point2D"
print(dist_to_origin(y))
```

Что выдаст этот код?

Методы

```
y <- c(1, 2, 3)
class(y) <- "Point2D"
print(dist_to_origin(y))
```

```
## [1] 2.236068
```

Никакой проверки, что то, что вы назвали объектом класса, этим классом является


ОДНА ОШИБКА
И ТЫ ОШИБСЯ
risovach.ru

Методы

```
print(x)
```

```
## [1] 1 2
## attr(,"class")
## [1] "Point2D"
```

```
print(y)
```

```
## [1] 1 2 3
## attr(,"class")
## [1] "Point2D"
```

Хотим сделать более красивый вывод

Методы

```
print.Point2D <- function(x) {
    desc <- paste("Point2D\n",
                  "x :", x[1], "\n",
                  "y :", x[2])
    cat(desc) # to print \n as newline
}
print.Point3D <- function(x) {
    desc <- paste("Point3D\n",
                  "x1:", x[1], "\n",
                  "x2:", x[2], "\n",
                  "x3:", x[3])
    cat(desc) # to print \n as newline
}
```

Методы

```
x <- c(1, 2); class(x) <- "Point2D"
y <- c(1, 2, 3); class(y) <- "Point3D"
print(x)
```

```
## Point2D
##   x : 1
##   y : 2
```

```
print(y)
```

```
## Point3D
##   x1: 1
##   x2: 2
##   x3: 3
```

print уже generic-функция

Что в R вы использовали и оно уже S3-объект?

А что еще в R вы использовали и оно уже S3-объект?

```
df <- head(starwars, 5)
print(class(df))
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```
print(typeof(df))
```

```
## [1] "list"
```

```
class(df) <- NULL
print(df)
```

```
## $name
## [1] "Luke Skywalker" "C-3PO"          "R2-D2"          "Darth Vader"
## [5] "Leia Organa"
##
## $height
## [1] 172 167  96 202 150
##
## $mass
## [1]  77  75  32 136  49
##
## $hair_color
## [1] "blond" NA      NA      "none"  "brown"
```

Статистические тесты

Понятие выборки

Генеральная совокупность (в англ. — population) — совокупность всех объектов (единиц), относительно которых учёный намерен делать выводы при изучении конкретной проблемы.

Выборка или выборочная совокупность — множество случаев (испытуемых, объектов, событий, образцов), с помощью определённой процедуры выбранных из генеральной совокупности для участия в исследовании.

Репрезентативность - выборка может рассматриваться в качестве репрезентативной или нерепрезентативной

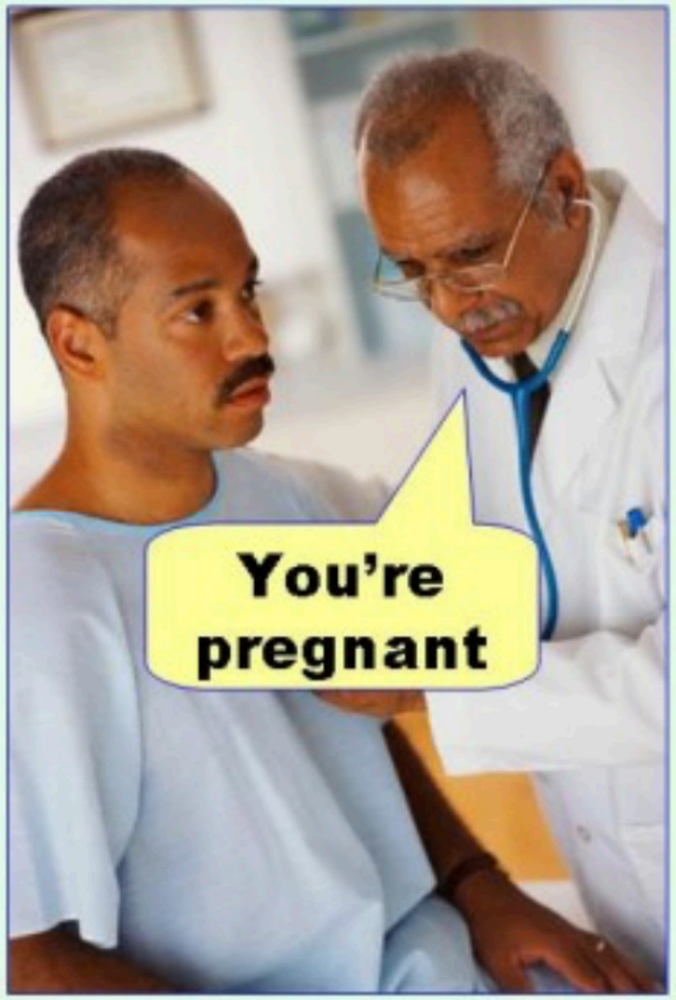
Тестирование гипотез

Нулевая гипотеза (H0) – это основное проверяемое предположение, которое обычно формулируется как отсутствие различий, отсутствие влияние фактора, отсутствие эффекта, равенство нулю значений выборочных характеристик и т.п.
Примером нулевой гипотезы в педагогике является утверждение о том, что различие в результатах выполнения двумя группами учащихся одной и той же контрольной работы вызвано лишь случайными причинами.

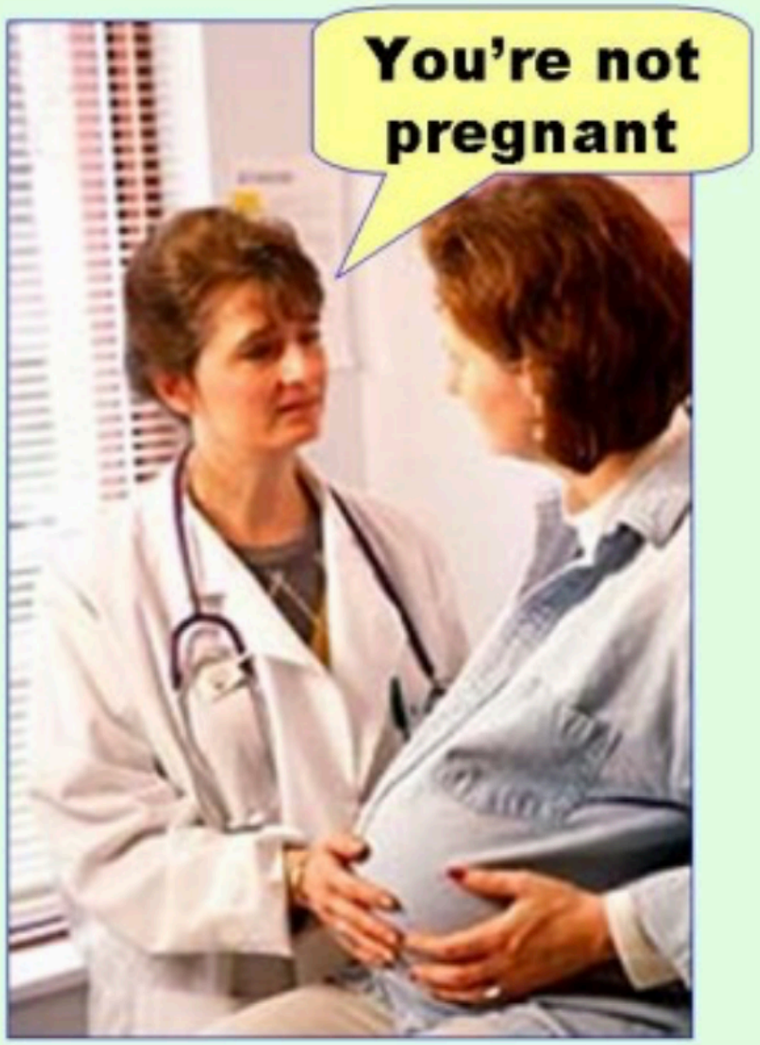
Другое проверяемое предположение (не всегда строго противоположное или обратное первому) называется конкурирующей или альтернативной гипотезой (H1). Обычно она соответствует предположению, что мы нашли значимое воздействие какого-то фактора

Ошибки первого и второго рода

Type I error
(false positive)



Type II error
(false negative)



Ошибки первого и второго рода

H0	верная	ложная
Отклоняется	Ошибка первого рода (alpha, FP)	Решение верное
Не отклоняется	Решение верно	Ошибка второго рода (beta, FN)

Задача

Представим, что мы хотим проверить, насколько хорошо витамин С помогает в лечении простуды. Для этого мы делим пациентов на пары (на основе пола, возраста, здоровья и т.д.). Далее считаем сколько, в скольких парах люди, принимавшие витамин С, выздоровели от простуды раньше. Гипотезы:

H0: P (витамин С лучше) = ½
H1: P (витамин С лучше) != ½

Допустим, что наш эксперимент состоял в том, что мы собрали 17 пар наблюдений и наблюдали в 13 парах, что раньше выздоровел принимавший витамин С.

Как оценить насколько подтвердилась наша гипотеза?

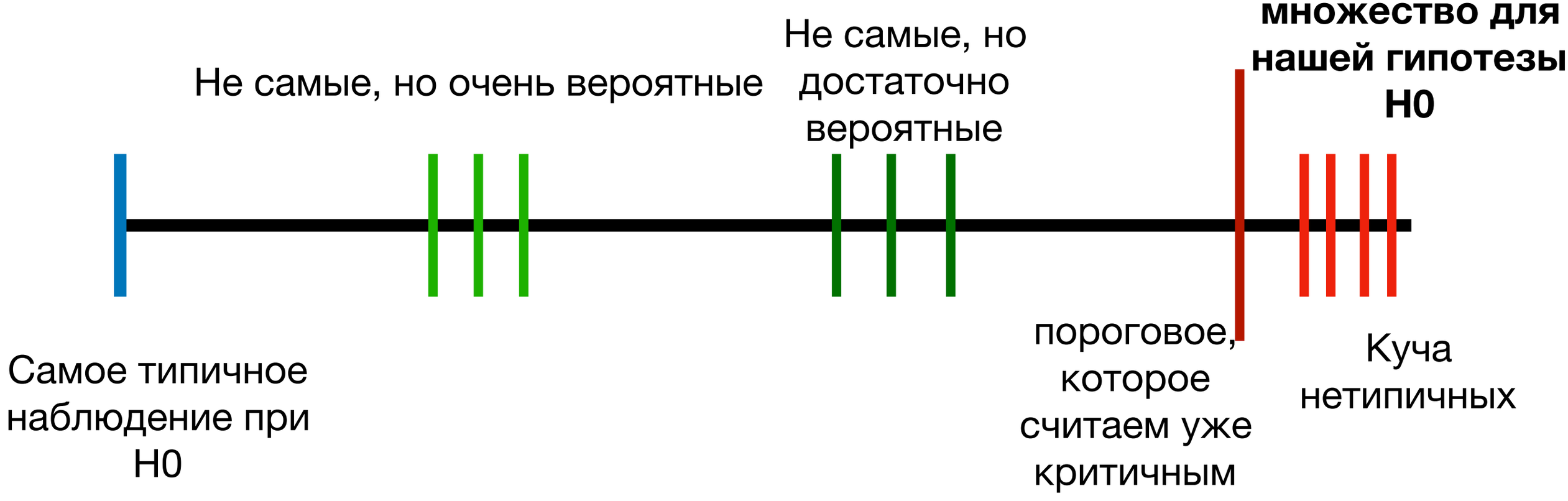
Задача

У нас есть 17 испытаний с вероятностью успеха p. По определению - биномиальное распределение.

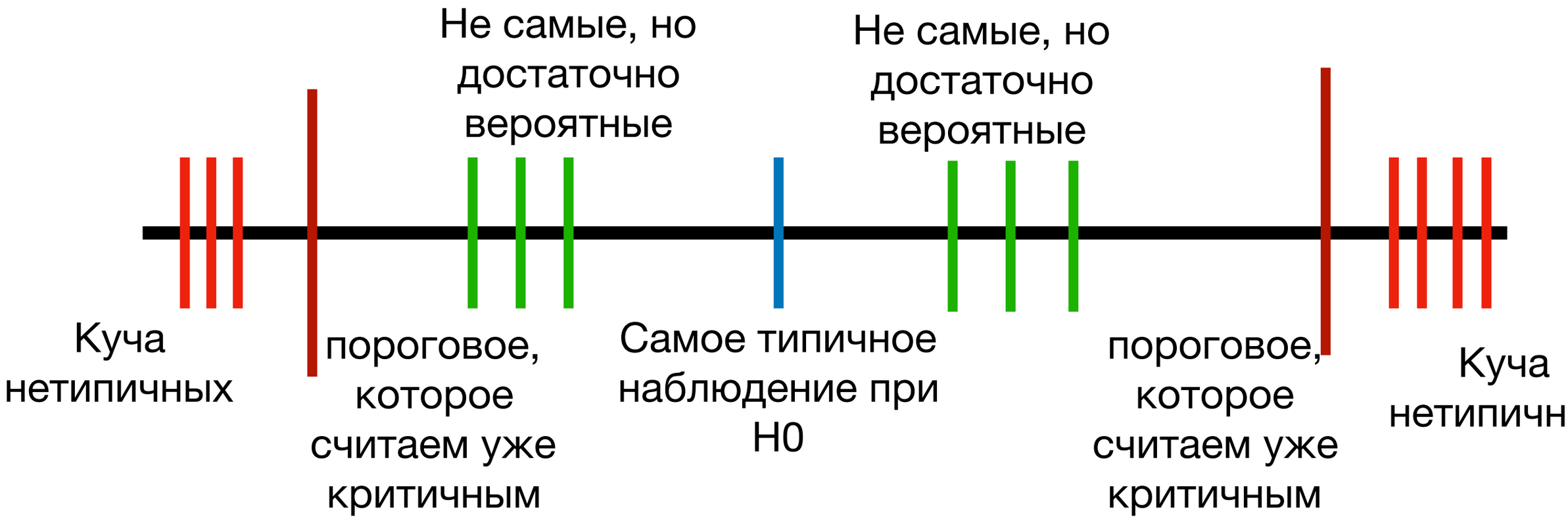
Можно посчитать вероятность нашего наблюдения при условии H0:

$$C_{17}^{13} p^{13} (1-p)^4 = 0.018$$

Но это не говорит нам о вероятности нашей гипотезы H0..
Можно рассуждать иначе - пусть мы будем считать это наблюдение значимым. Тогда логично считать и все менее вероятные при условии нашей гипотезы наблюдения значимыми.



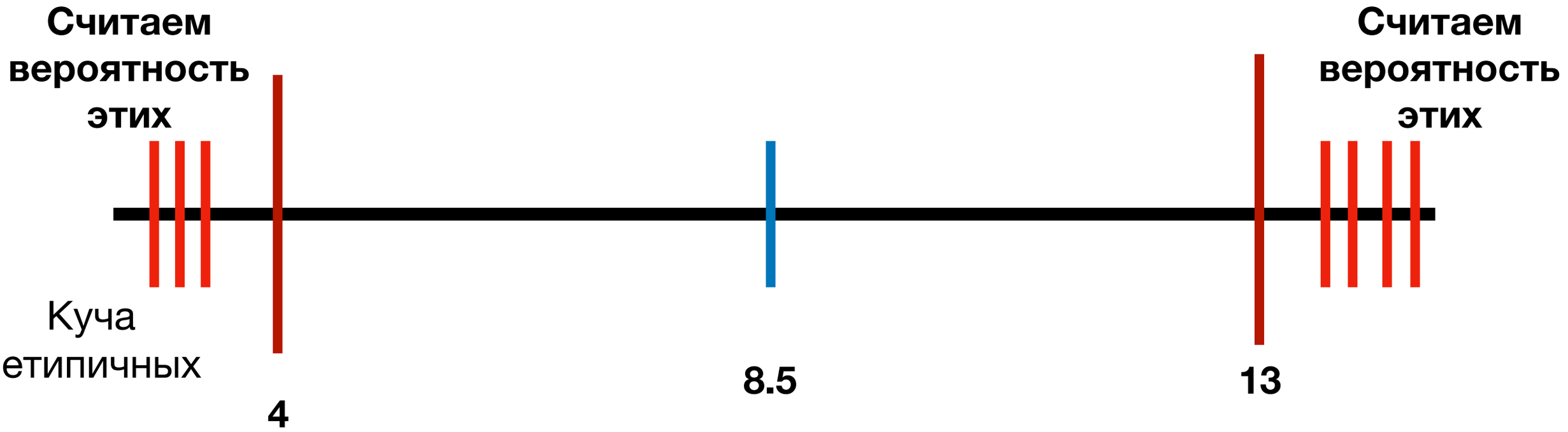
Задача



Часто у распределения нетипичные наблюдения с обеих сторон смотрим, потому иногда удобнее представлять что-то такое

Задача

Такие значения для нас - от 13 до 17, и от 4 до 0.



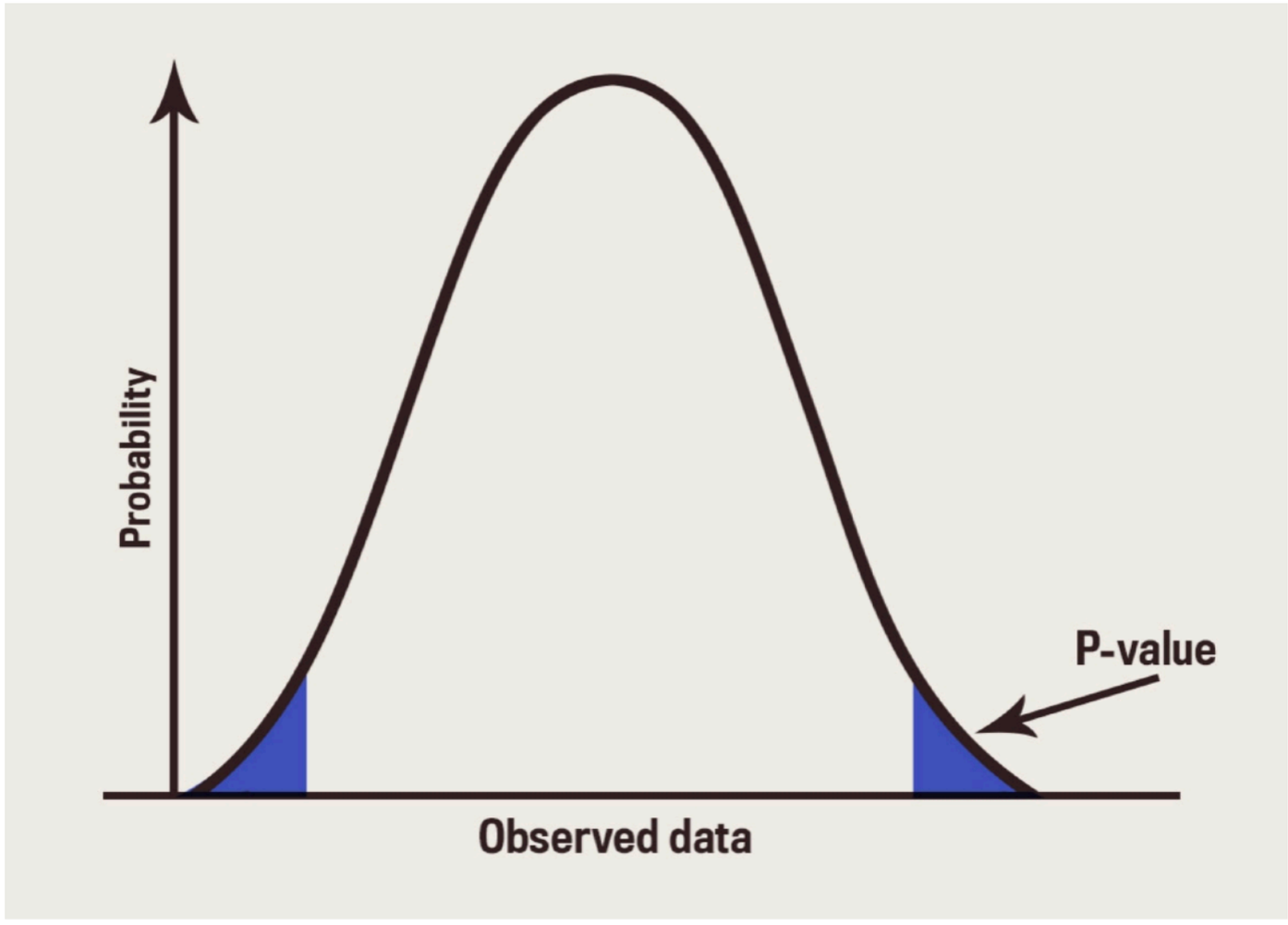
$$\sum_{i=13}^{17} C_{17}^{i} p^{i}(1-p)^{17-i} + \sum_{i=0}^{4} C_{17}^{i} p^{i}(1-p)^{17-i} = 0.049$$

Обычно берут порог 0.05, потому наше наблюдение значимо

P-value

P-value - вероятность получить результат как минимум такой же критический как тот, что мы наблюдаем, считая, что нулевая гипотеза является правильной

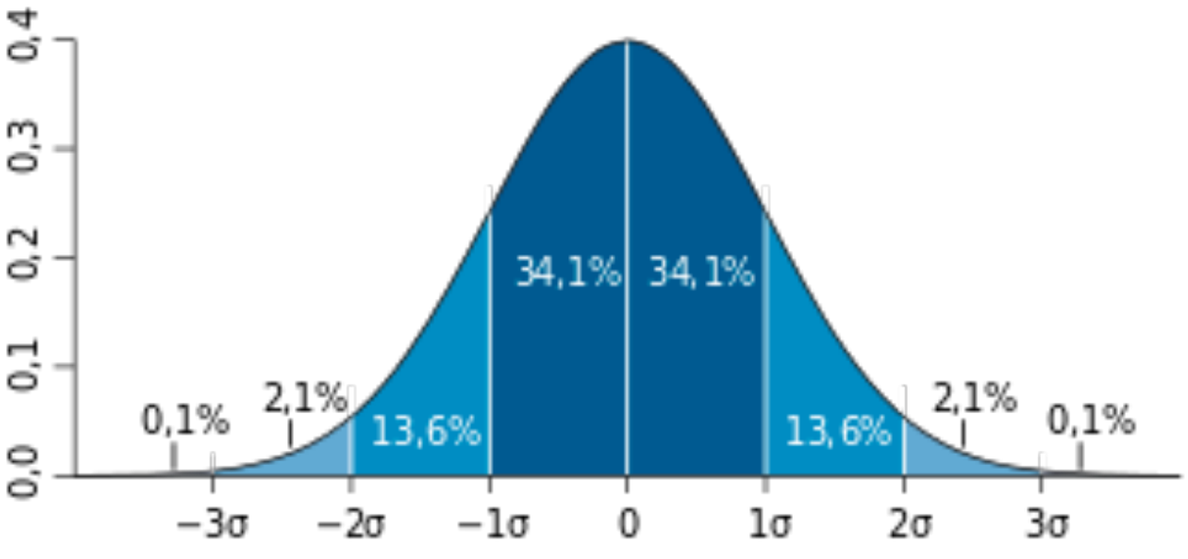
Другими словами - если нулевая гипотеза верна, то насколько вероятно получить ту выборку, которую мы получили, или более критичную



Тесты

Параметрические

Предполагают, что генеральная совокупность распределена по какому-то закону, использует параметры этой совокупности



Непараметрические

Не делает предположений о генеральной совокупности, критерий “свободен от распределения”.

Тест на равенство среднего

Какие знаете?

Одновыборочный тест Стьюдента

Если нет, то мы ее оцениваем выборочной дисперсией, и тогда тест Стьюдента

$$s^2 = \frac{\sum_i^N (X_i - \overline{X})^2}{N - 1}$$

$$\frac{\overline{X} - \mu}{s/\sqrt{N}} \sim t(df)$$

df - число степеней свободы.
В данном случае равно n - 1, так как для подсчета выборочного отклонения используется выборочное же среднее

R. t-test

Student's t-Test

Description

Performs one and two sample t-tests on vectors of data.

Usage

```
t.test(x, ...)

## Default S3 method:
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95, ...)
```

Все тесты Стьюдента в одном флаконе

Одновыборочный тест

```r
human_height <- starwars[starwars$species == "Human", ]$height
t.test(human_height, mu=180, alternative = "two.sided")
```

```
##
##  One Sample t-test
##
## data:  human_height
## t = -1.4899, df = 30, p-value = 0.1467
## alternative hypothesis: true mean is not equal to 180
## 95 percent confidence interval:
##  172.0466 181.2437
## sample estimates:
## mean of x
##  176.6452
```

Двухвыборочный тест

```
human_height <- starwars[starwars$species == "Human", ]$height
not_human_height <- starwars[starwars$species != "Human", ]$height
t.test(x=human_height, y=not_human_height, alternative = "two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  human_height and not_human_height
## t = 0.37878, df = 55.468, p-value = 0.7063
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -11.06723  16.22712
## sample estimates:
## mean of x mean of y
##  176.6452  174.0652
```

Важно!
В R из x вычитается y

Двухвыборочный тест

```
human_height_eyeblue <- starwars[starwars$species == "Human" &
                                 starwars$eye_color == "blue", ]$height
human_height_eyenotblue <- starwars[starwars$species == "Human" &
                                    starwars$eye_color != "blue", ]$height
t.test(x=human_height_eyeblue, y=human_height_eyenotblue,
       alternative = "two.sided", var.equal = T)
```

```
##
##  Two Sample t-test
##
## data:  human_height_eyeblue and human_height_eyenotblue
## t = -0.54474, df = 29, p-value = 0.5901
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -12.115584   7.019092
## sample estimates:
## mean of x mean of y
##  175.0833  177.6316
```

Двухвыборочный тест, левосторонняя альтернатива

```
human_height_eyeblue <- starwars[starwars$species == "Human" &
                                 starwars$eye_color == "blue", ]$height
human_height_eyenotblue <- starwars[starwars$species == "Human" &
                                    starwars$eye_color != "blue", ]$height
t.test(x=human_height_eyeblue, y=human_height_eyenotblue,
       alternative = "less", var.equal = T)
```

```
##
##  Two Sample t-test
##
## data:  human_height_eyeblue and human_height_eyenotblue
## t = -0.54474, df = 29, p-value = 0.295
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 5.400066
## sample estimates:
## mean of x mean of y
##  175.0833  177.6316
```

Двухвыборочный тест, правосторонняя альтернатива

```
human_height_eyeblue <- starwars[starwars$species == "Human" &
                                 starwars$eye_color == "blue", ]$height
human_height_eyenotblue <- starwars[starwars$species == "Human" &
                                    starwars$eye_color != "blue", ]$height
t.test(x=human_height_eyeblue, y=human_height_eyenotblue,
       alternative = "greater", var.equal = T)
```

```
##
##  Two Sample t-test
##
## data:  human_height_eyeblue and human_height_eyenotblue
## t = -0.54474, df = 29, p-value = 0.705
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -10.49656       Inf
## sample estimates:
## mean of x mean of y
##  175.0833  177.6316
```

Двухвыборочный тест, разница равна A

```
human_height_eyeblue <- starwars[starwars$species == "Human" &
                                   starwars$eye_color == "blue", ]$height
human_height_eyenotblue <- starwars[starwars$species == "Human" &
                                      starwars$eye_color != "blue", ]$height
t.test(x=human_height_eyeblue, y=human_height_eyenotblue, alternative = "two.sided", var.equal = T,
       mu = 10)
```

```
##
##  Two Sample t-test
##
## data:  human_height_eyeblue and human_height_eyenotblue
## t = -2.6825, df = 29, p-value = 0.01194
## alternative hypothesis: true difference in means is not equal to 10
## 95 percent confidence interval:
##  -12.115584   7.019092
## sample estimates:
## mean of x mean of y
##  175.0833  177.6316
```

Парный тест

```r
# Weight of the mice before treatment
before <-c(200.1, 190.9, 192.7, 213, 241.4, 196.9, 172.2, 185.5, 205.2, 193.7)
# Weight of the mice after treatment
after <-c(392.9, 393.2, 345.1, 393, 434, 427.9, 422, 383.9, 392.3, 352.2)

t.test(x=before, y=after, alternative = "two.sided", paired = T)
```

```
##
##  Paired t-test
##
## data:  before and after
## t = -20.883, df = 9, p-value = 6.2e-09
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -215.5581 -173.4219
## sample estimates:
## mean of the differences
##                  -194.49
```

R. z-test

```r
library("BSDA")
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'BSDA'
```

```
## The following object is masked from 'package:datasets':
##
##     Orange
```

```r
z.test(x=1:10, y=1:10, sigma.x = 1, sigma.y = 1)
```

```
##
##  Two-sample z-Test
##
## data:  1:10 and 1:10
## z = 0, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.8765225  0.8765225
## sample estimates:
## mean of x mean of y
##       5.5       5.5
```

Критерий Хи-квадрат

Тест на Goodness of fit

Насколько ваша модель распределения данной переменной описывает реально наблюдаемые значения

H0: модель верна

H1: Модель неверна

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

$$df = n - 1$$

n - число ячеек

Goodness of fit

- Всегда ли число степеней свободы n - 1?

Goodness of fit

- Всегда ли число степеней свободы n - 1?

Нет

Оно зависит от того, сколько условий вы накладываете

Goodness of fit

- Всегда ли число степеней свободы n - 1?

Нет

Оно зависит от того, сколько условий вы накладываете на ваши наблюдения/величин из них считаете непосредственно до теста

В предыдущем случае есть только одно условие - сумма всех наблюдений равна n.

Потому из числа наблюдений (n) мы и вычитаем 1

Сколько степеней свободы надо в случае:

- Проверить гипотезу о том, что наши наблюдения распределены по нормальному закону?

- Проверить гипотезу о том, что наши наблюдения распределены по равномерному закону?

- Проверить гипотезу о том, что наши наблюдения распределены по Пуассону с параметром 2?

Сколько степеней свободы надо в случае:

- Проверить гипотезу о том, что наши наблюдения распределены по нормальному закону?

  n - 1 - 2 = n - 3, считаем среднее и дисперсию
- Проверить гипотезу о том, что наши наблюдения распределены по равномерному закону?

  n - 1, мы это делаем по-умолчанию
- Проверить гипотезу о том, что наши наблюдения распределены по Пуассону с параметром 2?

  n - 1, мы взяли параметр не из наблюдений

Критерий Хи-квадрат

Тест на независимость

Используется как на то, есть ли значимая ассоциация между двумя факторными переменными

H0: факторы независимы

H1: факторы зависимы

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

$$df = (n-1) \cdot (m-1)$$

Где df - число степеней свободы, n - число разных значений первой переменной, m - число разных значений второй

chisq.test

Pearson's Chi-squared Test for Count Data

Description

`chisq.test` performs chi-squared contingency table tests and goodness-of-fit tests.

Usage

```
chisq.test(x, y = NULL, correct = TRUE,
           p = rep(1/length(x), length(x)), rescale.p = FALSE,
           simulate.p.value = FALSE, B = 2000)
```

Задача

Программист Петя считает, что количество лайков, которые соберут посты с шутками на тему неприятных особенностей языка, одинаковы. Для теста были выбраны языки C++, Python, Javascript, Java и R. Количество лайков для постов про эти языки составило соответственно:

17, 23, 72, 44, 65

Прав ли Петя? Уровень значимости 0.001, так он не хочет никого в случае чего обидеть незаслуженно.

```
x <-  c(17, 23, 72, 44, 65)
```

```
chisq.test(x=x, p = rep(1, length(x)) / length(x))
```

```
##
##  Chi-squared test for given probabilities
##
## data:  x
## X-squared = 54.181, df = 4, p-value = 4.823e-11
```

Задача

```
cont_tab <- rbind(c(40, 22, 17, 12),
                  c(15, 12, 20, 35),
                  c(35, 20, 22, 10))

chisq.test(x=cont_tab)
```

```
##
##  Pearson's Chi-squared test
##
## data:  cont_tab
## X-squared = 36.21, df = 6, p-value = 2.51e-06
```

Какая проблема с хи-квадрат тестом, реализованном в R?

chisq.test

Pearson's Chi-squared Test for Count Data

Description

`chisq.test` performs chi-squared contingency table tests and goodness-of-fit tests.

Usage

```
chisq.test(x, y = NULL, correct = TRUE,
           p = rep(1/length(x), length(x)), rescale.p = FALSE,
           simulate.p.value = FALSE, B = 2000)
```

Считаем частично руками, когда df не n-1

```
library(dplyr)
v <- starwars %>% filter(!is.na(species)) %>% group_by(homeworld) %>% summarize(s=sum(species == "Human"))
 %>% pull(s)
lambda <- mean(v)
p <- dpois(0:max(v), lambda = lambda)
p[length(p)] <- 1 - sum(p[1:(length(p) - 1)])

row <- table(factor(v, levels=0:max(v)) )

ch_t <- chisq.test(row, p = p) # incorrect, values < 5
```

```
## Warning in chisq.test(row, p = p): Chi-squared approximation may be incorrect
```

```
print(ch_t$statistic)
```

```
## X-squared
##  20147.46
```

```
print(1 - pchisq(ch_t$statistic, df=length(p) - 2))
```

```
## X-squared
##         0
```

Проблемы с критерием Хи-квадрат

Критерий Хи-квадрат можно применять только тогда, когда ожидаемое число наблюдений в каждой клетке больше 5.
Иначе необходимо использовать точный тест Фишера

Точный тест Фишера

Левый хвост, сложить вероятности всех таблиц здесь

Все хорошо

Правый хвост, сложить вероятности всех таблиц здесь

Таблица, перекошенная, как наша, но в другую сторону

Наша таблица

Таблицы с еще более перекошенной в другую сторону связью

	Исход есть	Исход а нет	Всего
Фактор есть	A	B	A+ B
Фактора нет	C	D	C + D
Всего	A + C	B + D	A + B + C + D

	Исход есть	Исход а нет	Всего
Фактор есть	A	B	A+ B
Фактора нет	C	D	C + D
Всего	A + C	B + D	A + B + C + D

Таблицы с еще более перекошенной в нашу сторону связью

Тест Фишера

fisher.test

Fisher's Exact Test for Count Data

Description

Performs Fisher's exact test for testing the null of independence of rows and columns in a contingency table with fixed marginals.

Usage

```
fisher.test(x, y = NULL, workspace = 200000, hybrid = FALSE,
            hybridPars = c(expect = 5, percent = 80, Emin = 1),
            control = list(), or = 1, alternative = "two.sided",
            conf.int = TRUE, conf.level = 0.95,
            simulate.p.value = FALSE, B = 2000)
```

fisher.test

```
cont_mat <- rbind(c(1, 9 ), c(11, 3))
fisher.test(cont_mat)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  cont_mat
## p-value = 0.002759
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.0006438284 0.4258840381
## sample estimates:
## odds ratio
## 0.03723312
```